**配置Linux环境、熟悉Linux实验报告**

姓名：徐昕妍 学号：2023K8009970008 提交日期：2024.9.21

一、摘要

本次实验旨在配置Linux环境，并通过一系列操作熟悉Linux系统的基本使用，掌握在Linux环境下进行文件管理、程序编译与运行、以及命令行工具的使用。实验采用实践操作的方式进行，主要步骤包括：在虚拟化平台上搭建Linux环境，使用终端命令创建目录、文件，编辑和编译C语言程序，执行Python脚本，以及使用base64命令对文件内容进行编码。实验过程中，重点练习了文件权限设置、环境变量配置、命令行参数传递等关键技术。

二、实验环境

本次实验所使用的基本环境为虚拟化平台VM Ware上搭建的Linux环境，发行版本为Ubuntu 22.04。不同于常用的Windows系统，Linux系统为一种开源操作系统，它基于UNIX操作系统。它是由芬兰学生林纳斯·托瓦兹（Linus Torvalds）于1991年发起并开发的。Linux系统的核心组件是内核，它负责管理系统的硬件资源，并提供基础的运行时环境给其他软件。

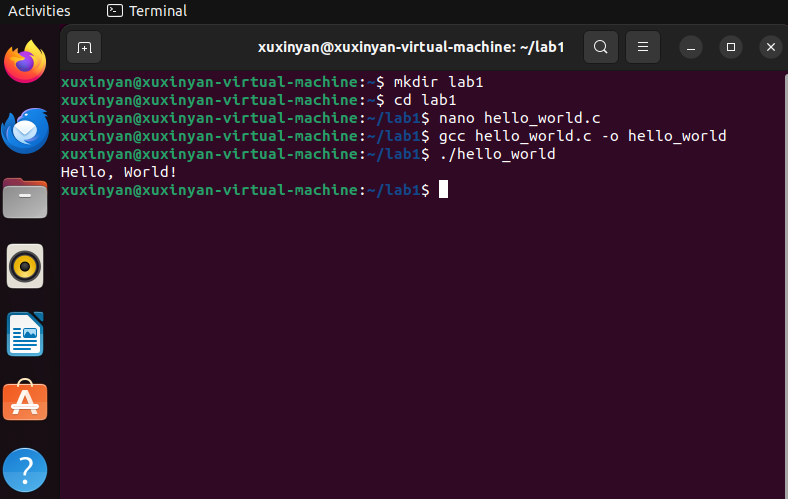
Linux的开源特性为开发者提供了极大的灵活性和透明性，用户可以自由查看、修改和分发Linux源代码。同时Linux系统以其稳定性和可靠性著称，而基于其开源特性，安全漏洞通常能被很快发现并修复。Linux还提供了强大的命令行界面，允许用户通过终端执行命令进行系统管理和日常任务，用户可以根据自己的需求定制Linux系统。

Linux系统的多样性和灵活性使其成为广泛用途的理想选择，从个人计算到企业级应用，再到科学研究，都能看到Linux的身影。

三、实验步骤

1、创建lab1目录。打开终端并执行命令 *mkdir lab1*，创建名为lab1的目录。

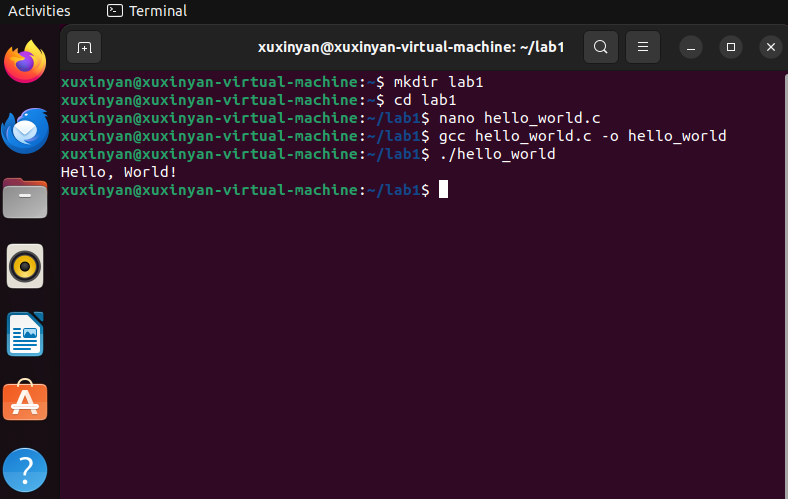
2、进入lab1目录。使用cd命令 *cd lab1* 进入lab1目录。执行上述命令后，当前工作目录变为lab1。

3、创建C源码，编译并运行hello，world！

（1）使用nano作为文本编辑器，输入命令 *nano hello\_world.c* 创建一个C源文件；

（2）在打开的编辑器中输入C语言代码，按ctrl+x，再按Y保存更改，最后按enter键保存文件；

（3）输入命令 *gcc hello\_world.c -o hello\_world* ，使用gcc编译器编译源文件；（4）输入命令 *./hello\_world* 运行编译生成的可执行文件；执行上述命令后，终端输出“Hello，World！”



4、用Python计算第100个斐波那契数并储存在文件中。

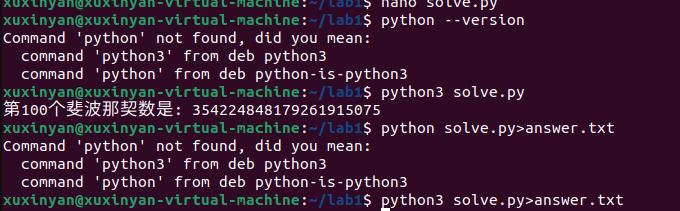
（1）使用nano作为文本编辑器，输入命令 *nano solve.py* ，创建文件；

（2）在编辑器中输入代码并保存文件；

（3）输入命令 *python --version* 查询Python版本；

（4）输入命令 *python3 solve.py* 运行文件；

（5）输入命令 *python3 solve.py>answer.txt* ,将输出重定向到answer.txt文件中。



值得注意的是，若直接输入实验文件中给出的命令*python solve.py>answer.txt* ，实际操作时则会因为没有标注python版本而报错。故为了确定电脑的Python版本，在（3）步使用*python --version*命令来查询Python版本。

1. 将/etc/os-release文件复制到lab1目录。由于lab1目录位于用户主目录下，故输入命令 *cp /etc/os-release ~/lab1/* ；若lab1位于当前工作目录下，则可以输入命令 *cp /etc/os-release ./lab1/* 。
2. 将/etc/fstab的base64编码写入文件。输入命令 *base64 /etc/fstab > fstab\_base64.txt* 。该命令代表“通过base64命令将/etc/fstab中的内容重定向到名为fstab\_base64.txt的文件中”。
3. 查看/proc/self/maps文件的内容并将结果保存在maps.txt中。输入命令 *cat /proc/self/maps > maps.txt* 。其中cat命令代表查看文件内容。
4. 将/bin/sh的十六进制数据储存到文件中。使用hexdump-C命令将将/bin/sh文件的十六进制数据储存到sh\_hexdump.txt文件中，输入命令 *hexdump -C /bin/sh > sh\_hexdump.txt* 。其中hexdump是用于显示文件十六进制和ASCII码的命令，-C是用于以可读的格式输出。
5. 返回上一级目录并打包lab1。
6. 使用*cd ..* 命令返回上一级目录；
7. 使用 *tar -czvf lab1-xuxinyan.tar.gz lab1* 命令来创建压缩文件。-c表示创建一个新的归档文件，-z表示通过gzip进行压缩；-v表示处理文件时显示详细信息，-f后跟随文件名，指定要创建的归档文件名。
8. 提交实验成果。

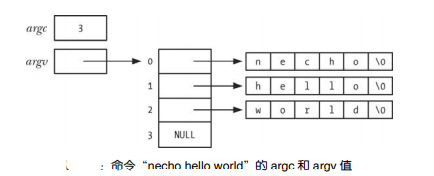
在实验中观察发现，大部分命令格式遵循“命令+文件1>文件2”意为通过命令将文件1进行操作，最终重定向到文件2。

1. 问题探究
2. 一个程序的命令行参数是什么？有什么限制？它在一个C程序中是如何使用的？
3. 命令行参数是指从命令行传递给程序的信息。在C语言中，这些参数通过main函数的参数列表传递给程序。main函数可以声明为以下两种形式之一来接收命令行参数：

int main(int argc, char \*argv[]);

int main(int argc, char \*\*argv);

其中，argc：表示参数的数量（argument count），类型为int；argv：是一个指向字符串的指针数组（argument vector），每个字符串都是一个参数，每一参数又都是以空字符（null）结尾的字符串，首个字符串 argv[0]——标识程序名本身，argv 中的指针列表以 NULL 指针结尾（即 argv[argc]为 NULL）。下图展示了程序所传入参 argc 和 argv 的数据结构。[[1]](#endnote-0)



1. 命令行参数主要存在以下三个限制：

数量限制：大多数操作系统对命令行参数的总长度有限制，这通常包括所有参数和它们之间的空格。例如，在Windows中，命令行长度通常限制为8191个字符。

参数长度限制：单个参数的长度也有限制，这通常由操作系统决定。例如，在Windows中，单个参数的长度通常限制为2048个字符。

参数类型：所有命令行参数都是以字符串的形式传递的。如果需要其他类型（如整数或浮点数），程序需要负责将字符串转换为所需的类型。

1. 在C程序中的使用：

以下面程序为例：

#include <stdio.h>

int main(int argc, char \*argv[]) {

// 打印参数的数量

printf("The program was called with %d arguments:\n", argc);

// 遍历并打印每个参数

for (int i = 0; i < argc; i++) {

printf("Argument %d: %s\n", i, argv[i]);

}

return 0;

}

当编译并运行这个程序时，我们可以从命令行传递任意数量的参数。例如：

./program\_name arg1 arg2 arg3

程序将输出：

The program was called with 4 arguments:

Argument 0: ./program\_name

Argument 1: arg1

Argument 2: arg2

Argument 3: arg3

2、Linux有哪些与文件相关的系统API？在C语言和Python语言中如何使用这些API？他们之间有什么关系？文件在Linux中的地位是什么？

（1）常用的文件系统API：

open(): 打开文件。

read(): 从文件中读取数据。

write(): 向文件写入数据。

close(): 关闭文件描述符。

creat(): 创建文件。

unlink(): 删除文件。

stat(): 获取文件状态信息。

fstat(): 获取打开文件的状态信息。

lstat(): 获取符号链接的状态信息。

chmod(): 改变文件的权限。

chown(): 改变文件的所有者。

truncate(): 截断文件到指定长度。

1. 在C语言中的应用：

open(“example.txt”,EXAMPLE);//打开文件

read(fd,buffer,sizeof(buffer));//读取文件

close（fd）；//关闭文件

在Python语言中的应用：

文件操作通常通过内置的open函数和文件对象的相应方法来完成，这些方法封装了底层的系统调用。如：

content=file.read()#读取文件内容

1. C语言和Python语言之间的关系：

Python的文件操作API在底层使用了C语言的系统调用；

Python提供了一个更高级、更易用的接口，隐藏了底层的细节；

在性能敏感的应用中，C语言直接使用系统调用可能会更高效。

1. 文件在Linux中的地位：

在Linux中，文件是数据和程序的基本存储单位，几乎所有资源都是以文件的形式存在。文件系统是Linux操作系统的核心组成部分，它管理着所有的文件和目录。

1. 综合应用——安装Docker并运行Hello World[[2]](#endnote-1)

1、检查卸载老版本docker。ubuntu下自带了docker的库，不需要添加新的源。  
但是ubuntu自带的docker版本太低，需要先卸载旧的再安装新的。输入命令为：

*apt-get remove docker docker-engine docker.io containerd runc*

但会存在显示无权限的情况，这时候需要使用root权限来卸载，即在命令前加上sudo。

2、更新软件包。在终端中执行以下命令来更新Ubuntu软件包列表和已安装软件的版本：*sudo apt update*

*sudo apt upgrade*

3、安装docker依赖。Docker在Ubuntu上依赖一些软件包。执行以下命令来安装这些依赖:*sudo apt-get install ca-certificates curl gnupg lsb-release*

4、添加Docker官方GPG密钥。执行以下命令来添加Docker官方的GPG密钥:

*sudo curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -*

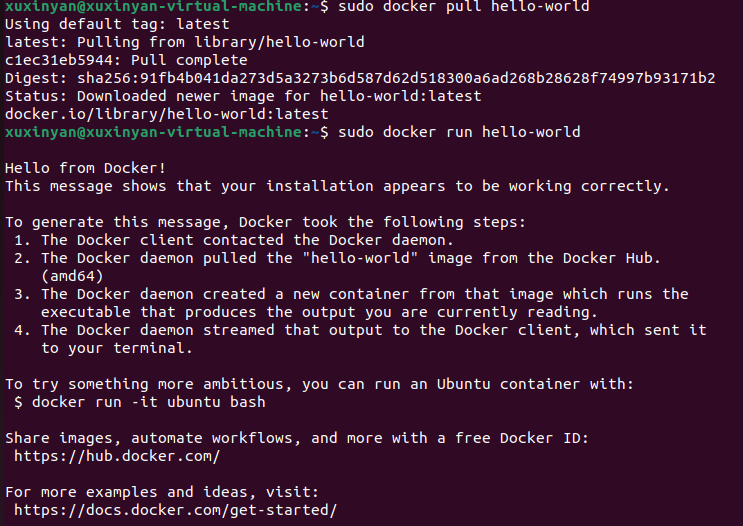
5、添加docker软件源。执行命令：*sudo add-apt-repository "deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu $(lsb\_release -cs) stable"*

6、安装docker。执行命令：*sudo apt-get install docker-ce docker-ce-cli containerd.io*

7、运行docker。通过启动docker来验证是否成功安装。命令如下：

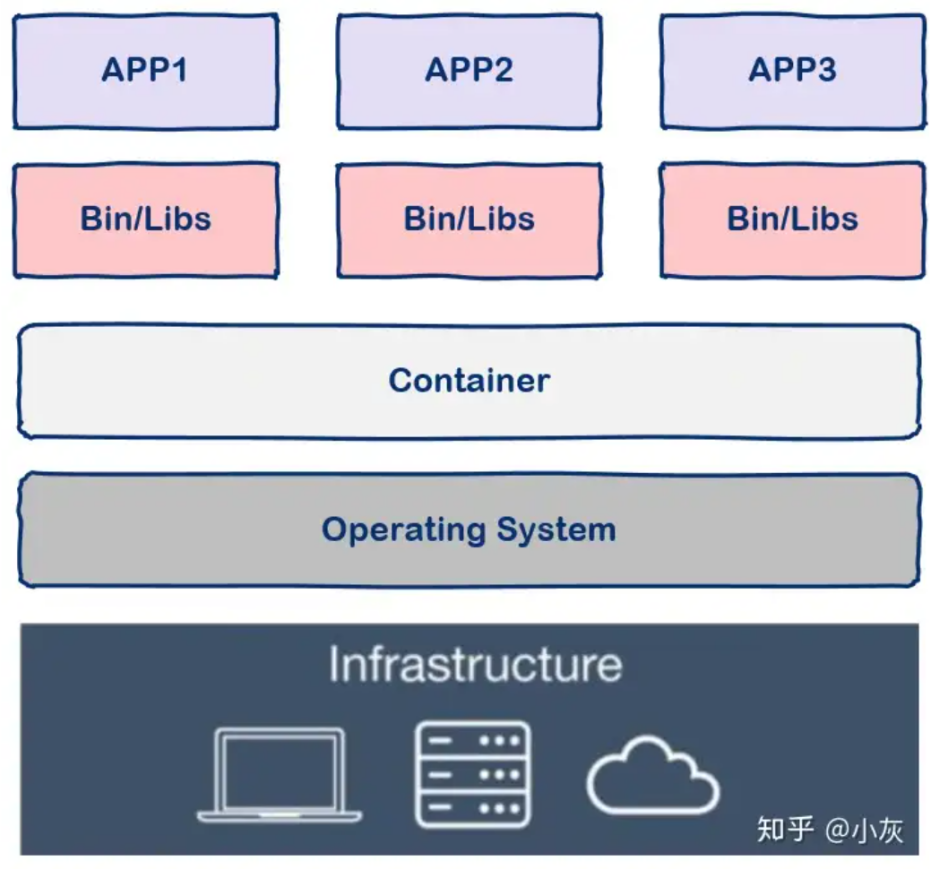
*sudo systemctl start docker*

1. 安装工具。执行命令：*sudo apt-get -y install apt-transport-https ca-certificates curl software-properties-common*
2. 重启docker。执行命令：s*udo service docker restart*
3. 抓取hello-world镜像。执行命令：*sudo docker pull hello-world*
4. 运行hello-world。执行命令：*sudo docker run hello-world*。得到结果：

**

对Docker的理解：

Docker是容器技术的一种实现。在理解docker之前，我们可以先理解容器技术。容器技术与虚拟机技术相似，都可以实现应用程序的隔离。与虚拟机通过操作系统实现隔离不同，容器技术只隔离应用程序的运行时环境但容器之间可以共享同一个操作系统，这里的运行时环境指的是程序运行依赖的各种库以及配置。



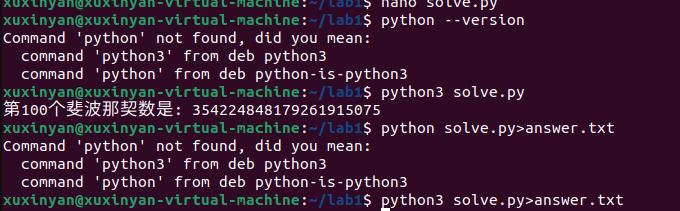
docker是一个用Go语言实现的[开源项目](https://zhida.zhihu.com/search?q=%E5%BC%80%E6%BA%90%E9%A1%B9%E7%9B%AE&zhida_source=entity&is_preview=1" \t "https://zhuanlan.zhihu.com/p/_blank)，可以让我们方便的创建和使用容器，docker将程序以及程序所有的依赖都打包到docker container，这样你的程序可以在任何环境都会有一致的表现。也就是说docker可以屏蔽环境差异。

此外docker的另一个好处就是快速部署，这是当前互联网公司最常见的一个应用场景，一个原因在于容器启动速度非常快，另一个原因在于只要确保一个容器中的程序正确运行，那么你就能确信无论在生产环境部署多少都能正确运行。[[3]](#endnote-2)

六、讨论与总结

在实验过程中，我遇到了以下问题：

1. 运行Python文件时命令错误。



如果依照实验文件中所给出的命令*python solve.py>answer.txt* 直接运行文件，则会出现报错。分析原因发现命令中必须写明Python版本。但如何确定电脑的Python版本又成为新的问题。经搜索查阅，我发现可以通过*python --version*命令来查询Python版本。因此在完成第四步时我加入了（3）步，并更改将命令更改为*python3 solve.py>answer.txt。*

1. 如何将虚拟机上的文件转存到本机上。

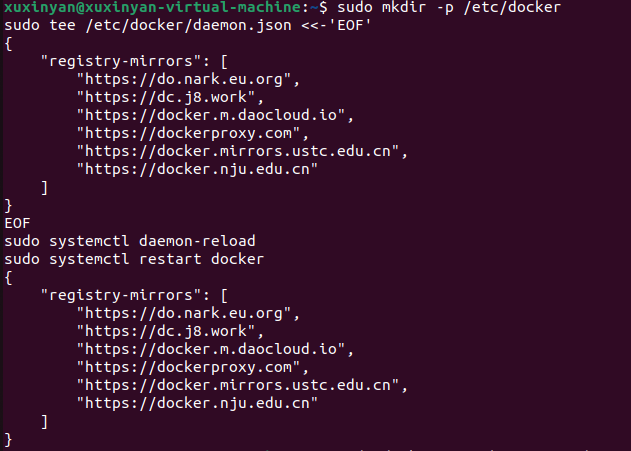
在提交实验报告时，为了方便，我希望将虚拟机上的文件转存到本机上并在本机上提交作业。经过搜索发现，主要可以通过一下几个途径实现：

1. 使用虚拟机上的共享文件夹。对于VM Ware，我们可以打开虚拟机设置，点击选项标签页，选择共享文件夹，点击总是启用，点击添加并选择本机上的文件夹，设置共享名等一系列操作完成。
2. 使用网络传输。当虚拟机和宿主机在同一网络中，我们可以通过在虚拟机中安装SSH服务，使用SCP或SFTP客户端从宿主机连接到虚拟机，并通过SCP或SFTP命令将文件从虚拟机传输到宿主机。
3. 使用虚拟机文件管理器。我们可以直接从宿主机的文件管理器访问虚拟机的文件系统。以VM Ware为例，在虚拟机的菜单中选择“VM” -> “映射虚拟机” -> “映射虚拟机文件夹”即可实现。
4. 借助物理介质，如USB驱动器。

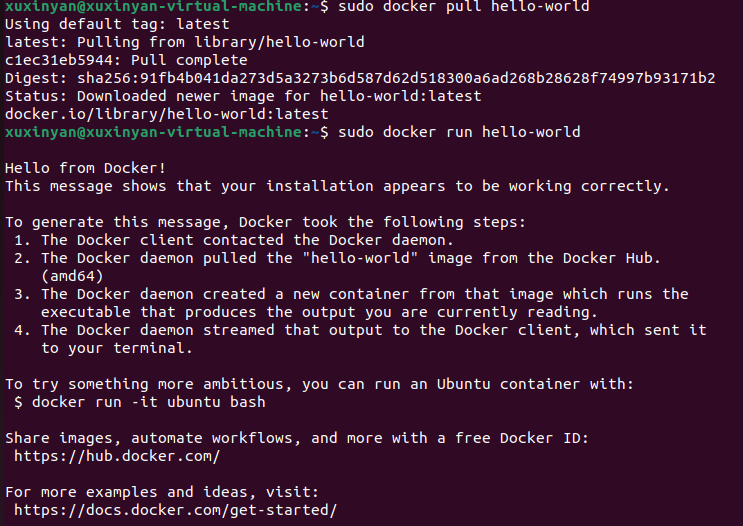
经本人探索与思考，我想到了另一种文件传输方法——微信文件传输助手。只需在Firefox上打开微信文件传输助手网页版，无需下载微信即可传输文件。该方法虽然朴实无华，但是操作难度极低，也不失为一种选择。

1. 从docker库中抓取镜像失败。

在进行抓取hello-world镜像时，反复出现了报错，经查证确认发现是连接超时。屏幕截图 2024-09-21 224251

搜索后发现可以通过配置加速地址的方法来加速，从而解决连接超时的问题。最开始我仅配置了一个镜像加速器，但依然会出现连接超时的报错。最终经过多次尝试，参考了网友北斗之缘[[4]](#endnote-3)提供的方法，配置多个镜像加速器，成功设置国内源。

随后再次进行抓取，抓取成功，成功运行hello-world。



总结：

本次实验过程中，我不仅成功搭建了Linux环境，还对Linux系统的基本命令、文件系统结构、权限管理、网络配置等方面进行了深入学习和实践。在实验中我不仅巩固了关于Linux的理论知识，而且在实际操作中培养了动手能力。通过对实验主题的深入探讨，我掌握了Linux环境的配置方法，并对系统管理有了更加直观的理解。同时面对出现的各种问题，我也逐渐掌握了解决问题的思路和方法。此次实验为我积累了宝贵的实践经验，对于今后在Linux平台上的学习和工作奠定了坚实的基础。

七、附录





1. OceanStar的学习笔记.C/C++编程：命令行参数.https://blog.csdn.net/zhizhengguan/article/details/109022694#:~:text=%E5%8F%AF%E4%BB%A5%E9%80%9A%E8%BF%87%E5%9C%A8%E7%A8%8B%E5%BA%8F%E7%9A%84ma [↑](#endnote-ref-0)
2. 软件测试大空翼.最详细的ubuntu安装docker教程.https://blog.csdn.net/Tester\_muller/article/details/131440306 [↑](#endnote-ref-1)
3. 程序员小灰.什么是docker？看这一篇干货文章就够了！.https://zhuanlan.zhihu.com/p/187505981 [↑](#endnote-ref-2)
4. 北斗之缘.从Docker拉取镜像一直失败超时？这些解决方案帮你解决烦恼.https://blog.csdn.net/weixin\_50160384/article/details/139861337#:~:text=Docker%20%E9%95%9C%E5%83%8F%E6%8B%89 [↑](#endnote-ref-3)